

AMENDMENTS TO THE CLAIMS

The listing of claims below replace all prior versions, and listings, of claims:

- A.I.
- 1 1. (Currently Amended) A method for use in a database system having plural
2 nodes, comprising:
3 storing a materialized join view based on at least two base relations;
4 storing at least one auxiliary relation containing one or more attributes of
5 one of the base relations, the auxiliary relation partitioned across the plural nodes
6 according to a join attribute; and
7 updating the at least one auxiliary relation in response to modification of
8 the one base relation.
 - 1 2. (Currently Amended) The method of claim 1, further comprising storing
2 the one base relation that is ~~not~~ partitioned across the plural nodes according to an
3 attribute that is different from the join attribute.
 - 1 3. (Original) The method of claim 2, further comprising:
2 receiving a tuple into the database system;
3 storing the tuple in one of the at least two base relations;
4 storing the tuple in the auxiliary relation; and
5 using the auxiliary relation to determine whether to update the
6 materialized join view.

1 4. (Currently Amended) A method comprising:

2 receiving a first tuple into a base relation at a first node of a parallel
3 database system having plural nodes, wherein the first tuple comprises a join attribute and
4 the base relation is ~~not~~ partitioned across the nodes according to an attribute different
5 from the join attribute;

6 storing the first tuple in an auxiliary relation at a second node of the
7 parallel database system, wherein the auxiliary relation is partitioned across the nodes of
8 the database system according to the join attribute;

9 identifying second tuples of a second relation;

10 joining the first tuple with the second tuples based on the join attribute to
11 produce join results; and

12 storing the join results in a join view.

1 5. (Cancelled)

1 6. (Cancelled)

1 7. (Currently Amended) The method of claim 4, wherein storing the first
2 tuple in an auxiliary relation at a second node comprises:

3 determining that a join view definition excludes an attribute of the first
4 tuple; and

5 not storing the excluded attribute in the auxiliary relation.

AI 1 8. (Currently Amended) The method of claim 4, ~~wherein storing the tuple in~~
2 ~~an auxiliary relation at a second node comprises~~ further comprising:
3 receiving a third tuple into the base relation;
4 determining that a join view definition includes a condition on one of the
5 attributes of the third tuple; and
6 ~~identifying the attribute in the tuple;~~
7 determining that the condition is not met by the one of the attributes of the
8 third tuple; and
9 not storing the second tuple in the auxiliary relation.

1 9. - 12. (Cancelled)

1 13. (Currently Amended) A database system comprising:
2 a storage modules ~~module~~ to store base relations and at least ~~one~~ a first
3 auxiliary relation corresponding to a first one of the base relations, the ~~at least one~~ first
4 auxiliary relation containing one or more attributes of the ~~one~~ first base relation, the ~~at~~
5 ~~least one~~ first auxiliary relation partitioned across the storage modules differently than the
6 ~~one~~ first base relation, the storage ~~module~~ modules further to store a join view based on a
7 join of the base relations; and
8 a controller adapted to update the join view using ~~the~~ at least the first ~~one~~
9 auxiliary relation.

1 14. (Currently Amended) The database system of claim 13, wherein the
2 controller is further adapted to receive a tuple and store the tuple in ~~one of the~~ first base
3 relation ~~relations~~ and in ~~one of the at least one~~ the first auxiliary ~~relations~~ relation.

1 15. (Original) The database system of claim 14, wherein the controller is
2 further adapted to not update the join view after receiving some tuples.

1 16. (Currently Amended) An article comprising a medium storing instructions
2 for enabling a processor-based system having plural nodes to:

3 store a join view to store join results of a join of at least first and second
4 base relations based on a join condition including a first attribute of the first base relation
5 and a second attribute of the second base relation;

6 receive a first tuple into a the first base relation at a first node, wherein the
7 first tuple comprises ~~a join~~ the first attribute and the first base relation is ~~not~~ partitioned
8 across the plural nodes according to an attribute other than the join first attribute;

9 store the first tuple in ~~an~~ a first auxiliary relation at a second node,
10 wherein the first auxiliary relation is partitioned across the plural nodes according to the
11 ~~join~~ first attribute;

12 identify second tuples of a the second base relation; and

13 join the first tuple with the second tuples to produce join results for
14 updating the join view; and

15 ~~store the join results in a join view.~~

1 17. (Currently Amended) The article of claim 16, further storing instructions
2 for enabling the processor-based system to:

3 ~~identify second join attributes in the second tuples; and~~

4 compare the second ~~join~~ attributes of the second tuples with the ~~join~~ first
5 attribute of the first tuple to produce the join results for updating the join view ~~relation~~.

1 18. (Currently Amended) The article of claim 16, further storing instructions
2 for enabling the processor-based system to:

3 determine that a join view definition excludes an attribute of the first
4 tuple; and

5 not store the excluded attribute in the first auxiliary relation.

1 19. (Currently Amended) The article of claim 16, further storing instructions
2 for enabling the processor-based system to:

A³¹
4 determine that a join view definition includes a condition on one of the
attributes of the first base relation ~~tuple~~; and

5 identify the one attribute in ~~the~~ a received third tuple;

6 determine that the condition is not met by the received third tuple; and

7 not store the received third tuple in the first auxiliary relation.

1 20. - 23. (Cancelled)

A²
1 24. (New) The method of claim 1, wherein storing the materialized join view
2 comprises storing a materialized join view containing tuples derived from a join of the at
3 least two base relations.

1 25. (New) The method of claim 24, wherein the one of the base relations
2 comprises a first base relation, the method further comprising:

3 partitioning the first base relation across the plural nodes in a first way;

4 and

5 partitioning the auxiliary relation across the plural nodes in a second,
6 different way according to the join attribute.

1 26. (New) The method of claim 25, further comprising:

2 receiving a first tuple to insert into the one of the base relations;

3 inserting the first tuple into the first base relation; and

4 inserting at least a portion of the first tuple into the auxiliary relation.

1 27. (New) The method of claim 26, further comprising:
2 distributing the first tuple from a first one of the plural nodes to a second
3 one of the plural nodes, wherein inserting the first tuple into the auxiliary relation is
4 performed at the second node; and
5 joining the first tuple with at least a second tuple associated with a second
6 one of the base relations in the second node.

1 28. (New) The method of claim 27, wherein inserting the first tuple into the
2 first base relation is performed at the first node, wherein joining the first tuple with at
3 least the second tuple is performed at the second node instead of the first node.

1 29. (New) The method of claim 28, further comprising storing a second
2 auxiliary relation containing one or more attributes of the second one of the base
3 relations, the second auxiliary relation partitioned according to a join attribute.

1 30. (New) The method of claim 29, wherein joining the first tuple with the
2 second tuple comprises joining the first tuple with the second tuple contained in the
3 second auxiliary relation.

1 31. (New) The method of claim 30, further comprising using a result of the
2 join of the first tuple with the second tuple in the second auxiliary relation to update the
3 materialized join view.

1 32. (New) The method of claim 1, further comprising maintaining an index on
2 the auxiliary relation.

1 33. (New) The method of claim 4, wherein joining the first tuple with the
2 second tuples is performed at the second node.

1 34. (New) The database system of claim 13, further comprising plural nodes,
2 the storage modules in respective plural nodes, and the controller comprises plural
3 processors in respective nodes.

A2
1 35. (New) The database system of claim 34, wherein a first one of the nodes is
2 adapted to receive a first tuple,
3 the processor in the first node to insert the first tuple into the first base
4 relation, and
5 the processor in a second one of the nodes to insert the first tuple into the
6 first auxiliary relation.

1 36. (New) The database system of claim 35, wherein the first node is adapted
2 to distribute the first tuple to the second node.

1 37. (New) The database system of claim 36, the storage modules in the second
2 node to store a portion of a second auxiliary relation to store one or more attributes of a
3 second one of the base relations, the storage module in the second node to store a portion
4 of the join view that is distributed across the plural storage modules, and the processor in
5 the second node to join the first tuple with tuples in the portion of the second auxiliary
6 relation to update the portion of the join view in the storage module of the second node.

1 38. (New) The database system of claim 37, the join view to store join results
2 based on a join condition including at least a first attribute of the first base relation and a
3 second attribute of the second base relation, wherein the first auxiliary relation is
4 partitioned across the storage modules according to the first attribute, and the second
5 auxiliary relation is partitioned across the storage modules according to the second
6 attribute.

1 39. (New) The database system of claim 38, the storage modules to store a
2 first index on the first auxiliary relation, and the storage modules to store a second index
3 on the second auxiliary relation.

1 40. (New) The database system of claim 13, the join view to store results of a
2 join of the base relations based on a join condition and selection condition, the controller
3 to store tuples of the first base relation that satisfy the selection condition in the first
4 auxiliary relation, and the controller to not store tuples of the first base relation that do
5 not satisfy the selection condition in the first auxiliary relation.

A2
1 41. (New) The database system of claim 13, the join view to store results of a
2 join of the base relations based on a query containing a select clause and a join condition,
3 the select clause specifying one or more attributes of the first base relation,
4 the controller to store the one or more attributes of the first base relation
5 specified by the select clause in the first auxiliary relation, and the controller to not store
6 other attributes of the first base relation not specified by the select clause in the first
7 auxiliary relation.

1 42. (New) The database system of claim 13, the join view to store results of a
2 join of at least the first base relation and a second base relation based on a join condition
3 including a first attribute of the first base relation and a second attribute of the second
4 base relation, wherein the first attribute is a key of the first base relation and the second
5 attribute is a foreign key of the second base relation that references the first attribute,
6 the controller to, in response to detecting that the first attribute is a key of
7 the first base relation and that the second attribute is a foreign key of the second base
8 relation that references the first attribute, create the first auxiliary relation to store the one
9 or more tuples of the first base relation but to not create a second auxiliary relation to
10 store tuples of the second base relation.

1 43. (New) The article of claim 16, wherein the processor-based system
2 comprises plural storage modules, the first base relation partitioned across the storage
3 modules according to the attribute other than the first attribute, and
4 the first auxiliary relation partitioned across the storage modules according
5 to the first attribute.

1 44. (New) The article of claim 43, wherein the plural nodes contain the
2 storage modules and plural processors, wherein the instructions when executed cause the
3 processor-based system to further distribute the first tuple from the first node to the
4 second node.

A2
1 45. (New) The article of claim 44, wherein the instructions when executed
2 cause the processor-based system to further:
3 store a second auxiliary relation containing one or more tuples of the
4 second base relation;
5 partition the second auxiliary relation across the storage modules
6 according to the second attribute; and
7 partition the second base relation across the storage modules according to
8 an attribute of the second base relation other than the second attribute.

1 46. (New) The article of claim 45, wherein identifying the tuples of the second
2 base relation comprises identifying the tuples of the second auxiliary base relation, and
3 wherein joining the first tuple with the second tuples comprises joining the first tuple
4 with the second tuples of the second auxiliary relation.

1 47. (New) The article of claim 16, wherein the join view stores join results of
2 the join of the at least first and second base relations based on a query containing the join
3 condition, wherein the instructions when executed cause the processor-based system to
4 further determine whether the query specifies one or more elements that enable storage of
5 less than the entire first base relation in the first auxiliary relation.

1 48. (New) The article of claim 47, wherein the first auxiliary relation stores
2 the entire first base relation in response to determining the one or more elements do not
3 exist.

1 49. (New) The article of claim 48, wherein the one or more elements comprise
2 a selection condition in a WHERE clause of the query.

A2
1 50. (New) The article of claim 48, wherein the one or more elements comprise
2 less than all of the attributes of the first base relation specified by a select clause in the
3 query.
